Graph Adaptive Semantic Transfer for Cross-domain Sentiment Classification

Kai Zhang
Anhui Province Key Lab. of Big Data Analysis and Application, University of S&T of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
kkzhang0808@mail.ustc.edu.cn

Qi Liu, Zhenya Huang
Anhui Province Key Lab. of Big Data Analysis and Application, University of S&T of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
{qiliuql,huangzhy}@ustc.edu.cn

Mingyue Cheng
Anhui Province Key Lab. of Big Data Analysis and Application, University of S&T of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
mycheng@mail.ustc.edu.cn

Kun Zhang
School of Computer Science and Information Engineering, Hefei University of Technology
Hefei, China
zhang1024kun@gmail.com

Mengdi Zhang, Wei Wu
Meituan
Beijing, China
mdzhangmd@gmail.com
wuwei19850318@gmail.com

Enhong Chen*
Anhui Province Key Lab. of Big Data Analysis and Application, University of S&T of China
Hefei, China
cheneh@ustc.edu.cn

ABSTRACT

Cross-domain sentiment classification (CDSC) aims to use the transferable semantics learned from the source domain to predict the sentiment of reviews in the unlabeled target domain. Existing studies in this task attach more attention to the sequence modeling of sentences while largely ignoring the rich domain-invariant semantics embedded in graph structures (i.e., the part-of-speech tags and dependency relations). As an important aspect of exploring characteristics of language comprehension, adaptive graph representations have played an essential role in recent years. To this end, in the paper, we aim to explore the possibility of learning invariant semantic features from graph-like structures in CDSC. Specifically, we present ***G**raph **A**daptive **S**emantic **T**ransfer (**GAST**)* model, an adaptive syntactic graph embedding method that is able to learn domain-invariant semantics from both word sequences and syntactic graphs. More specifically, we first raise a *POS-Transformer* module to extract sequential semantic features from the word sequences as well as the part-of-speech tags. Then, we design a *Hybrid Graph Attention (HGAT)* module to generate syntax-based semantic features by considering the transferable dependency relations. Finally, we devise an *Integrated aDaptive Strategy (IDS)* to guide the joint learning process of both modules. Extensive experiments on four public datasets indicate that GAST achieves comparable effectiveness to a range of state-of-the-art models.

*Corresponding Author.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531984

CCS CONCEPTS

• **Information systems** → **Retrieval tasks and goals**; **Sentiment analysis**.

KEYWORDS

Text Mining; Sentiment Analysis; Domain Adaptation; Graph Embedding; Web Content Analysis

ACM Reference Format:
Kai Zhang, Qi Liu, Zhenya Huang, Mingyue Cheng, Kun Zhang, Mengdi Zhang, Wei Wu, Enhong Chen. 2022. Graph Adaptive Semantic Transfer for Cross-domain Sentiment Classification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3531984

1 INTRODUCTION

Sentiment classification is a fundamental task in natural language processing (NLP). Over the past decades, many supervised machine learning methods such as logistic regression, support vector machines, and neural networks [17, 27, 33, 48] are applied to the task. However, due to the domain shift problem, directly using the off-the-shelf sentiment classifiers to a new domain (e.g., dataset) may lead to a significant performance drop [28].

To address the problem, cross-domain sentiment classification (CDSC), which refers to utilizing the valuable knowledge in the source domain to help sentiment prediction in a target domain, has been proposed and extensively studied in the last decade. In the literature, many previous methods focus on learning the shared features, i.e., common sentiment words [1, 6, 32], part-of-speech tags [42] and syntactic tree [49], with traditional machine learning methods, which are usually based on handcrafted features and fail to model the deep semantic representations across domains because the universal structures are essentially human-curated and expensive to acquire across domains.

Subsequently, with the great progress of deep neural networks in numerous NLP tasks, some scholars explore deep models to learn latent representations of domain-shared information. Most of these studies [10, 11] focus on extracting features from the word sequences and embed the sequential features into deep semantic representations through various methods, e.g., memory network [26], recurrent neural network [3, 44], attention mechanism [25, 47], and the large-scale pre-trained models [9, 23]. However, these studies only concentrate on modeling the domain-invariant semantics from textual word sequences, while largely discarding the exploration of adaptive graph syntactic information, i.e., the part-of-speech tags (POS-tags) and dependency relations.

Actually, as an important aspect of exploring the characteristics of language comprehension, syntactic information exploration has made significant progress, especially being combined with graph-based models in many NLP tasks [18, 22, 41]. For example, in ABSA, using syntactic information to enhance the semantic representation of aspects has become the basic configuration of the SOTA model [37, 40]. However, current advanced methods of CDSC learn semantics only from standardized word sequences while largely ignoring those adaptive syntactic structure information. Therefore, despite their popularity, efforts to incorporate universal language structure correspondence between domains such as part-of-speech tags and dependency relations into the domain adaptation framework have been sporadic. To this end, we identify multiple advantages of using adaptive syntactic-semantic for domain adaptation.

First, sentiment words play a crucial role in CDSC [42], while POS tags can distinguish sentiment words (e.g., *"horrible"* and *"interesting"* in Figure 1) via the POS tag "JJ" in a natural way, i.e., the "JJ" label means the word is an adjective. Unfortunately, recent studies only explore word semantics from the pre-trained word embeddings, which may not be sufficient to identify sentiment words in the CSDC task. **Second**, the sentiment polarity of reviews is largely influenced by the sentiment word's neighbors, whether they are in-domain or across-domain. As Figure 1 (b) shows, the neighbor *"quite"* is more important than non-adjacent words (e.g.,*"the"* and *"book"*) for the sentiment word *"interesting"*. Meanwhile, different neighbors' syntactic relations also have different influences for each word. For instance, the neighbor word *"quite"* also plays a more critical role than neighbor *"was"* for sentiment word *"interesting"*. Thus, existing methods that solely rely on sequential relations may lead to sub-optimal sentiment prediction in CDSC. **Third**, as shown in Figure 1 (c), the syntactic graph structures of sentences in different domains are remarkably similar, which means that the syntactic rules are domain-invariant and can be naturally transferred across domains. However, the exploration of these crucial features is still neglected, and how to train a graph adaptive semantic transferable model for CDSC has not been fully considered.

Following the above intuitions, in this paper, we propose a ***G**raph **A**daptive **S**emantic **T**ransfer (**GAST**)* model, which aims to learn textual semantics and graph adaptive semantics for cross-domain sentiment classification. Generally, GAST improves the semantic representation and transferable knowledge between domains by

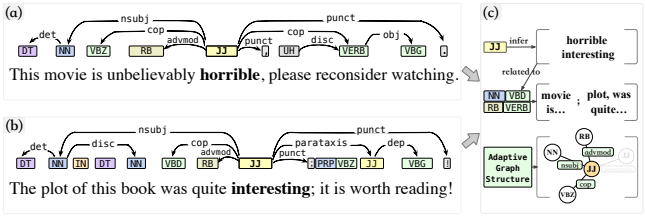


Figure 1: The transferable syntactic structures[*] of two examples (i.e., (a), (b)). The colorful boxes ("DT") and black lines (e.g., "det") indicate POS tags and syntactic relations, respectively. As shown in (c), the syntactic structures are similar between domains so that it is easy for human to understand the hidden knowledge behind sentences in different domains. However, those adaptive graph features are largely ignored by existing domain adaptation research.

aggregating the information from both word sequences and syntactic graphs. Specifically, GAST mainly contains two modules to learn comprehensive semantics. The first one is *POS-based Transformer (POS-Transformer)*, which includes a new multi-head attention mechanism to encode the word semantics with the help of POS tags. The other is *Hybrid Graph Attention (HGAT)*, which aims to learn and weigh the semantic influence between words and their neighbors with the help of the domain-invariant dependency relations. Finally, we propose a novel *Integrated aDaptive Strategy (IDS)* which integrates an adversarial training and pseudo-label based semi-supervised learning to distill transferable semantic features. Extensive experiments on real-world datasets demonstrate the effectiveness of our proposed approach. In summary, the contributions of this work can be summarized as:

- To the best of our knowledge, we present the first solution to address the CDSC problem by incorporating domain-invariant semantic knowledge from word sequences and syntactic graph structures simultaneously.
- We propose a novel *Graph Adaptive Semantic Transfer (GAST)* model for syntactic graph learning. GAST contains a *POS-Transformer* to learn sequential semantic from word sequences and POS tags, and a *HGAT* to fully exploit adaptive syntactic knowledge with the help of dependency relations.
- We further design an integrated adaptive strategy to optimize the transferability of the GAST model. Experimental results show that the proposed model achieves better results compared to other strong competitors.

2 RELATED WORK

In the following, we will introduce two research topics which are highly relevant to this work, i.e., cross-domain sentiment classification and syntax modeling in NLP tasks.

2.1 Cross-domain Sentiment Classification

Cross-domain sentiment classification (CDSC) aims to generalize a classifier that is trained on a source domain into a target domain in which labeled data is scarce. In the field of CDSC, a group of methods focuses on exploiting the explicit domain-shared knowledge [30, 32, 42]. Among them, Blitzer et al. [1, 2] proposed and extended the structural correspondence learning to identify the domain-shared features from different domains. Xia and Zong [42] designed an

[*]The syntactic structure of the sentences are constructed by the Stanford CoreNLP toolkit: https://stanfordnlp.github.io/CoreNLP/.

ensemble model to learn common features by using the part-of-speech tags. These earlier methods need to select domain-shared features manually while obtaining rich artificial features is a time-consuming and expensive process.

Recent years, many researchers studied the problem [6, 11, 13, 25, 44, 45] through the deep neural networks. Among them, Glorot et al. [13] first proposed a deep learning model named Stacking Denoising Autoencoder (SDA), which aimed to improve the scalability of high-dimensional data. Later, Chen et al. [6] extended it as marginalized Stacked Denoising Autoencoder (mSDA). Along this line, Yu and Jiang [44] leveraged two auxiliary tasks to learn in-depth features together with a CNN-based classifier. Ganin et al. [11] introduced a general domain adaptation strategy of the task by applying a gradient reversal layer. Ghosal et al. [12] enriched the semantics of a document by exploring the role of external commonsense knowledge. Li et al. [25, 26] incorporated the adversarial memory network and hierarchical attention transfer network into domain-adversarial learning to automatically identify invariant features. Zhang et al. [47] designed an interactive transfer network, which aims to extract interactive relations between sentences and aspects. Du et al. [9] proposed a BERT-DAAT model and a post-training procedure to enforce the model to be domain-aware. Despite the promising results, most of these methods process sentences as whole word sequences and ignore the domain-invariant syntactic structures of the sentence.

Although some earlier studies have studied the syntactic information in CDSC [30, 42], they only focused on modeling POS tags while largely ignoring the graph adaptive semantics behind the dependency relations. Thus, in this paper, we design a graph adaptive semantic transfer model to learn comprehensive semantics from both word sequences and syntactic structures.

2.2 Syntax Modeling in NLP Tasks

Syntactic information has been verified to be essential for many NLP tasks, such as aspect-based sentiment analysis (ABSA) [18, 35, 36, 40, 46]. Among those methods, Huang et al. [18] utilized the syntactic information to represent the sentence as a graph structure instead of a word sequence. Wang et al. [40] reshaped the dependency tree to learn aspect-aware semantics from syntactic information. These methods have indicated that syntactic information positively affects semantic representation. However, they are designed explicitly for aspect-based sentiment or other tasks, which can not perfectly apply to the CDSC task. For example, although numerous methods use syntactic information to help sentiment prediction in ABSA, this syntactic information supports the model to understand semantics rather than transfer semantics better. Thus, the performance of those approaches drops a lot under the domain adaptation scenario.

Furthermore, there are also some syntax-based studies in cross-domain aspect and opinion co-extraction task [7, 22, 41]. However, most of those methods utilized the dependency relationships to generate an auxiliary label [7] of the sentence or generate an auxiliary task for relation classification [41], which is not in line with the purpose of CDSC. Though these models are not suitable for CDSC, they effectively prove the effectiveness of syntactic information in cross-domain tasks. Therefore, in this paper, we leverage syntactic structures as transferable information and incorporate it into the domain adaptation framework to learn domain-invariant features.

However, though remarkable advance has been gained in the task, the above methods can only handle the sequential semantics in word sequences instead of the rich syntactic structure knowledge the sentence contains, which may not be able to learn transferable syntactic knowledge that is important to human. Unlike previous methods, our GAST can explore transferable semantics from syntactic structures and sequential information, thus can better simulate human's syntax rules.

3 METHODOLOGY

3.1 Problem Definition

Suppose we have two domains, $\mathcal{D}_s$ is the source domain (i.e., contains labeled data $\{x_s^i, y_s^i\}_{i=1}^{n_{sl}} \in \mathcal{D}_s^l$ and unlabeled data $\{x_s^i\}_{i=n_{sl}+1}^{n_s} \in \mathcal{D}_s^u$) and $\mathcal{D}_t = \{x_t^i\}_{i=1}^{n_t}$ is the unlabeled target domain. x_*^*, y_*^* denote samples and the corresponding label. The notations n_{sl}, n_s and n_t are the number of labeled data in source domain, the number of data in source domain and the number of data in target domain, respectively. Our goal is to learn a robust classifier from samples in the source and target domain and adapt it to predict the sentiment polarity of unlabeled examples in the target domain.

3.2 Syntactic Graph and Embedding

In the task, each input sentence s contains n words marked as $s = \{s_1, s_2, ..., s_n\}$. To learn a syntax-aware representation, we transform each sentence into a syntactic dependency tree T using an off-the-shelf dependency parser[†] [8]. Note that, the dependency tree can be represented as a syntax graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V}$ includes all words of the sentence, $\mathcal{A}$ is adjacent matrix with $A_{ij} = 1$ if there exists a dependency relation between word s_i and s_j, and $A_{ij} = 0$ otherwise. $\mathcal{R}$ is a set of syntactic relations (e.g., *det*, *nsubj* and *cop*), where R_{ij} corresponds to the relation label of A_{ij}.

For our model, we conduct two types of word embedding methods: the GloVe [34] and pre-trained BERT embedding. Specifically, for GloVe embeddings, we map each word over a GloVe word matrix to get the vector. For BERT embeddings, we use an off-the-shelf toolkit[‡] to encode each word and obtain the embeddings for each word instead of GloVe. For simplicity, we leverage $v_i \in \mathbb{R}^d$ as the representation vector of the i-th word in sentence, and the original word sequence s is transformed into an embedding matrix, i.e., $E \in \mathbb{R}^{n \times d}$. Moreover, we encode each word's POS tag to an embedding vector $t_i \in \mathbb{R}^{d_t}$ and transform each syntactic relation label R_{ij} to a vector $r_{ij} \in \mathbb{R}^{d_r}$, where d, d_t and d_r are the dimension of different embedding spaces.

3.3 POS-Transformer

Since the superiority of the Transformer model in various sequential tasks [38], we utilize transformer-based encoder to learn semantic knowledge from the word sequence as well as the POS tags. As shown in Figure 2 (a), the basic of our POS-Transformer is a new multi-head attention that creatively incorporates POS tags along with the traditional word sequences.

To be specific, for each attention head $i \in [1, I]$, we project the word's embedding matrix E into the query, key, and value matrices,

[†]https://github.com/allenai/allennlp.
[‡]https://bert-as-service.readthedocs.io/en/latest/.

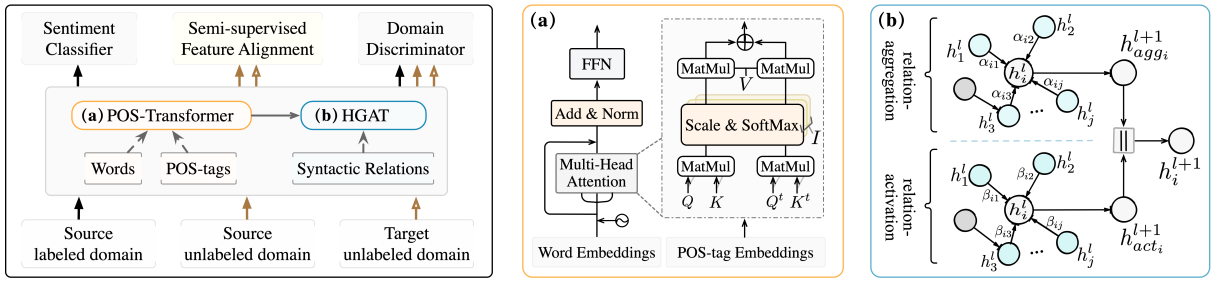


Figure 2: The architecture of GAST, which consists three parts: (a) the *POS-Transformer* that can learn sequential semantic representation by considering both the word sequences and POS tags; (b) the *HGAT* module which can exploit adaptive syntactic semantics of the sentence through the syntactic relation graph. (c) an *IDS* (i.e., Sentiment Classifier, Semi-supervised Feature Alignment and Domain Discriminator) to optimize the model and encourage it to be domain-invariant and syntax-aware.

denoted as Q_i, K_i, V_i. Apart from word's embeddings, we also map the whole tag embedding matrix E^t into matrices Q_i^t and K_i^t, while keeping the value V_i in this part to explore the interactive influence between POS tags and the words. Then, we incorporate external POS tags knowledge along with word's sequential information to learn POS-based semantic representation (i.e., Z) of the sentence:

$$Z = concat(z_1, z_2, \ldots, z_I), \tag{1}$$

$$z_i = Att.(Q_i, K_i, V_i) + Att.(Q_i^t, K_i^t, V_i), \tag{2}$$

$$Att.(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d/I}}\right)V, \tag{3}$$

where I denotes the number of attention head in transformer, $Att.()$ is the attention function. After getting the deep latent representation Z, we apply non-linear transformations on it and get the final output feature R of the POS-Transformer module:

$$R = \max(0, ZW_1 + b_1)W_2 + b_2, \tag{4}$$

where W_1, W_2, b_1, b_2 are the weight and bias parameters. Through the above procedure, the POS-Transformer can calculate the correlation between each word with the help of POS tags. Therefore, the sequential semantic knowledge hidden in the word sequences and POS tags can be fully extracted.

3.4 Hybrid Graph Attention

In order to effectively learn syntactic graph embedding with full consideration of the syntactic dependencies, we adopt GAT to learn the relational features of the sentence. Generally, given a word w_i (i.e., node i) with its deep hidden representation h_i^l at l-th layer. GAT updates the node's hidden state (i.e., h_i^{l+1}) at l+1 layer by calculating a weighted sum of its neighbor states through the masked attention (i.e., compute w_j for nodes $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is the neighborhood of node i in the syntactic graph).

However, the vanilla GAT uses an adjacent matrix as structural information, thus omitting dependency relations. Unlike vanilla GAT, we design a new Hybrid GAT to enhance information exchange among words via syntactic relations. As Figure 2 (b) shows, HGAT contains two different calculation methods for better relation representation (i.e., relation-aggregation and relation-activation). The first is the relation-aggregate function, which is designed to learn transferable syntactic relations during the aggregate process. Thus, we could conduct the following formulas:

$$h_{agg_i}^{l+1} = \|_{k=1}^{\bar{K}} \sigma\Big(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{lk} W_{lk} h_j^l\Big), \tag{5}$$

$$f_{ij}^{lk} = \sigma(a_{lk}^T [W_{lk} h_i^l \| W_{lk} h_j^l \| W_{lk} r_{ij}]), \tag{6}$$

$$\alpha_{ij}^{lk} = \frac{\exp\left(f_{ij}^{lk}\right)}{\sum_{j=1}^{\mathcal{N}_i} \exp\left(f_{ij}^{lk}\right)}, \tag{7}$$

where $\|$ denotes the concatenation of vectors, W_{lk} is a learnable transformation matrix and $\bar{K}$ is the number of attention head in GAT. Besides, a_{lk}^T is a learnable parameter and α_{ij}^{lk} is the attention coefficient in the k-th head at l-th layer. $\sigma()$ is the *LeakyReLU* function. r_{ij} represents the syntactic relation embedding between word i and word j.

As shown in Equation 6, the above relation-aggregate function can learn syntactic features by splicing the representation of both syntactic relation and nodes. However, the splicing operation is relatively intuitive and straightforward, which may not be able to capture the interactive influence between nodes and their syntactic dependency relations. To this end, we calculate the activation probability of each syntactic dependency relation by leveraging the scaled dot-product attention [38] so that the adaptive impact of different syntactic relations can be reassigned and further explored. Specifically, in our implementation, the relation-activation function is represented as:

$$\beta_{ij}^{lk} = \frac{\exp\left(F_{act.}(h_i^l, h_j^l)\right)}{\sum_{j=1}^{\mathcal{N}_i} \exp\left(F_{act.}(h_i^l, h_j^l)\right)}, \tag{8}$$

$$F_{act.} = \frac{\left(W_Q^{lk} h_i^l\right)\left(W_K^{lk} h_j^l + W_{Kr}^l r_{ij}\right)^T}{\sqrt{d/\bar{K}}}, \tag{9}$$

$$h_{act_i}^{l+1} = \|_{k=1}^{\bar{K}} \sigma\Big(\sum_{j \in \mathcal{N}_i} \beta_{ij}^{lk} (W_V^{lk} h_j^l + W_{Vr}^l r_{ij})\Big), \tag{10}$$

where W_Q^{lk}, W_K^{lk}, W_V^{lk}, W_{Kr}^{l} and W_{Vr}^{l} are learnable parameter matrices which are shared across attention heads. With the relation-attentional function, the weight β_{ij}^{lk} greatly enhances the explanatory ability of the model and enables GAST to learn crucial features from the neighbors with high relation scores in the syntactic graph.

Through the above two relational functions, we can obtain two syntax-enhanced word representations. In order to better improve the comprehensiveness of syntactic information representation, we generate the final representation of each word at l+1 layer through:

$$h_i^{l+1} = h_{agg_i}^{l+1} \parallel h_{act_i}^{l+1}. \tag{11}$$

The sentence's final representation H is the average pooling result of each word representation in the sentence. Since HGAT is able to calculate the correlation between neighbor words in the adaptive syntactic graph, the transferable semantic features contained in the syntactic structures can be fully learned.

3.5 Integrated Adaptive Strategy

In this subsection, we introduce how GAST obtains the syntax-aware and domain-invariant features through an integrated adaptive strategy, which mainly includes three loss components. As shown in Figure 2, the strategy includes three loss functions: a classifier loss for sentiment knowledge learning, a discriminator loss for invariant feature extracting across domains, as well as a syntax feature alignment loss for syntax-aware feature alignment.

Sentiment Classifier. The sentiment classifier is simply defined as $\hat{y}_s = softmax(W_s H + b_s)$, which is used to classify sentiment polarities. The objective loss function of the classifier is defined as:

$$L_c = -\frac{1}{n_s^l} \sum_{i=1}^{n_s^l} (y_s^i \ln \hat{y}_s^i + (1 - y_s^i) \ln(1 - \hat{y}_s^i)), \tag{12}$$

where y_s^i is the ground-truth for the i-th sample in the source labeled domain $\mathcal{D}_s^l$. W_s and b_s represent learnable parameters. The classifier loss L_c will train the model to learn better sentiment-aware features from both sequential information and syntactic structures.

Domain Discriminator. Cross-domain sentiment classifiers perform well when the learned features are domain-invariant. To this end, we devise a domain discriminator, which aims to facilitate knowledge transfer between domains. Specifically, we feed the feature H into the softmax layer for domain classification $\hat{y}_d = softmax(W_d H + b_d)$. The optimization goal is to train a model that can fool the discriminator so that the learned features can be transferred from domain-specific to domain-invariant [21, 36]. Therefore, we reverse the domain label in the training process and the optimization objective function can be defined as:

$$L_d = -\frac{1}{N} \sum_{i=1}^{N} (y_d^i \ln \hat{y}_d^i + (1 - y_d^i) \ln(1 - \hat{y}_d^i)), \tag{13}$$

where y_d^i is the reversed ground-truth of sample i (i.e., swap the sample's domain label). N is the sum of n_s from the source domain and n_t is comes from the target domain.

Semi-supervised Learning (SSL). Since there are a large amount of unlabeled data in the source and target domain, it is hard to train an optimal classifier. Fortunately, syntactic information plays a critical role in semantic representation which is helpful for sentiment classification [18, 40]. Thus, we utilize the syntactic information of unlabeled data in both domains to estimate the sentiment label so that GAST can eliminate the sentiment discrepancies between domains (i.e., explore and align sentiment features via syntactic-semantic information) via semi-supervised learning [15].

Specifically, we first attempt to estimate the sentiment "pseudo" label through semi-supervised learning. Inspired by [24], we minimize the entropy penalty to disambiguate the positive and negative samples over the unlabeled data from both domains. The loss function is:

$$L_a = -\frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{C} \tilde{y}^i \ln \tilde{y}^i , \tag{14}$$

where $\tilde{y}^i$ is the label distribution estimated by our model, C is the number of categories and M is the sum of $n_s - n_{sl}$ and n_t. Note that, the domain discrepancy can be effectively reduced through feature alignment [15, 24].

3.6 Model Training

Since there are three objective functions in the model, we conduct an integrated strategy to jointly optimize the final loss. The formulation is defined as:

$$L = \lambda_c L_c + \lambda_d L_d + \lambda_a L_a , \tag{15}$$

where λ_c, λ_d and λ_a are hyper-parameters to balance different objective losses. The training goal is to minimize the integrated loss L with respect to the model parameters. Additionally, all the parameters are optimized by the standard back-propagation algorithm.

3.7 Summary and Remarks

It should be noted that in this paper we focus on learning domain-invariant semantics from both sequential texts and adaptive syntactic structures simultaneously. We optimize the model with an integrated adaptive strategy, which deeply explores the impacts of the adaptive graph structures for cross-domain sentiment classification. Although there have been some studies (e.g., in the ABSA task [14, 31, 40, 46]) to utilize syntactic graph structures to enhance semantic representation, the transferability exploration for adaptive graph structure information is still limited. That is, our proposals are the first solution to learn domain adaptive graph semantics for CDSC, and we encourage more effective studies to be explored and further improve our graph adaptive framework.

4 EXPERIMENTS

4.1 Dataset Setup

We evaluate GAST on four widely-used Amazon datasets, i.e., DVD (***D***), Book (***B***), Electronics (***E***) and Kitchen (***K***). As shown in Table 1, for each domain, there are 2,000 labeled reviews (i.e., 1,000 positive and 1,000 negative) as well as 4,000 unlabeled reviews. We follow the dataset configurations as previous studies [9, 25, 26], that is, we randomly choose 800 positive and 800 negative reviews from the source domain as training data, the rest 400 as validation data to find the best hyperparameters, and all labeled reviews from the target domain for testing.

Table 1: Statistics of datasets after pre-processing.

Domains	Testing set percentage			
	#Train	#Vali.	#Test	#Unlabel
Books	1,600	400	2,000	4,000
DVD	1,600	400	2,000	4,000
Electronics	1,600	400	2,000	4,000
Kitchen	1,600	400	2,000	4,000

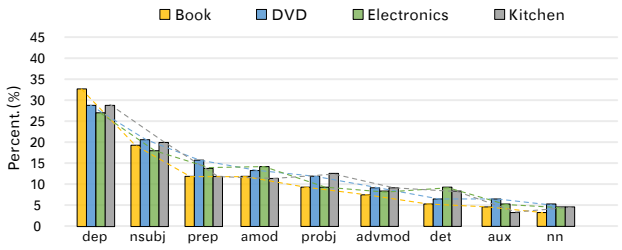


Figure 3: The percent of transferable dependency relations in different domains. We visualized the top 9 relations.

4.2 Data Analysis

In this subsection, we count the ratio of different syntactic relationships as illustrated in Figure 3. We can observe some phenomena intuitively. First, for each syntactic dependency relation, the proportions between various domains are close, meaning each sentence's components might be remarkably similar, even in different domains. Besides, the curve (i.e., dotted lines) of the syntactic relations is identical for each domain. The statistical observations above may indicate that the syntactic structures of the sentence are domain-invariant between domains from the perspective of data mining.

4.3 Baseline Methods

We compare the GAST model with multiple representative transfer baselines as well as some non-transfer approaches, which have achieved significant performance in recent years. The methods are listed below.

- **SCL** [2] is a linear method, which aims to solve feature the mismatch problem by aligning domain common and domain unique features.
- **SFA** [32] is a method which aims to build a bridge between the source and the target domains by aligning common and unique features.
- **mSDA** [6] is proposed to automatically learn a unified feature representation for sentences from a large amount of data in all the domains.
- **DANN** [6] is based on adversarial training. DANN performs domain adaptation with the representation encoded in a 5000-dimension feature vector.
- **AMN** [26] is a method which learns domain-shared features based on memory network sentiment.
- **HATN** [25] is a hierarchical attention transfer network which is designed to focus on both of the word-level and sentence-level sentiment.
- **IATN** [47] is an interactive attention transfer model which focus on mining the deep interactions between the context words and the aspects.
- **BERT-DAAT** [9] contains a post-training and an adversarial training process which aims to inject target domain knowledge to BERT and encourage it to be domain-aware.

Besides, we also borrow three competitive non-transfer deep representation learning method for comparison, i.e., Naive LSTM [16], TextGCN [43] and FastGCN [5]. The detail is illustrated as follow.

- **LSTM** [16] utilizes neural network to learn the hidden states and obtain the averaged vector through mean pooling to predict the sentiment polarity.
- **TextGCN** [43] a simple and effective graph neural network for text classification that captures high-order neighborhoods information from the syntactic graph.
- **FastGCN** [5] a fast improvement of the GCN model for learning graph embeddings. Here, we perform it on our syntactic graph to learn syntax-aware sentiment features.

4.4 Implementation Details

In the experiments, we initialize the dimension of the GloVe embeddings to 300 and utilize the BERT-base uncased model (layer=12, head=12, hidden=768) [19] in the off-the-shelf toolkit. The dimension of POS tags (d_t) and syntactic relations (d_r) is initialized to 30. The number of attention heads I and $\bar{K}$ are set to 8 and 3. We adopt Adam [20] as optimizer with learning rate 10^{-4} and dropout rate as 0.25 to train the model with the batch size of 32. The final settings of λ_c, λ_d and λ_a are 1, 1 and 0.8, respectively. All of the methods are implemented by Python and are trained on a Linux server with two 2.20 GHz Intel Xeon E5-2650 CPUs and four Tesla V100 GPUs. Finally, follow most previous studies [9, 25, 47], we use the widely used metric (i.e., accuracy) for the model evaluation. We will release the code and dataset once the paper is accepted.

4.5 Experimental Results

In this section, we evaluate the performance of our model on public datasets along with detailed analysis of results in Table 2. The major results are summarized as follows:

- In most tasks, neural-based transfer models (e.g., DANN and IATN) perform better compared with the SCL and SFA, which only manually selects common features (i.e., "pivots"). The phenomenon demonstrates the power of deep methods, especially the models specially designed for CDSC. Meantime, the graph-based models outperform the sequential model LSTM and some transferable models (e.g., SFA and DANN) a lot, proving that the graphical syntactic structure is important for cross-domain semantic representation.
- We also observe that the performance of GAST outperforms most baseline methods. Specifically, comparing with those neural-based transfer models (e.g., AMN, HATN, and IATN), our GAST model outperforms most of them. We conjecture one possible reason is that the performance of GAST can be significantly improved by fully exploring graphical syntactic structures of the sentences, i.e., POS-tags and syntactic dependency relations. Besides, the comparison with

Table 2: Sentiment classification accuracy (%) on the twelve transfer tasks.

Baselines	*DVD* (*D*)			*Book* (*B*)			*Electronics* (*E*)			*Kitchen* (*K*)		
	D↦*B*	*D*↦*E*	*D*↦*K*	*B*↦*D*	*B*↦*E*	*B*↦*K*	*E*↦*D*	*E*↦*B*	*E*↦*K*	*K*↦*D*	*K*↦*B*	*K*↦*E*
SCL	77.8	75.2	75.5	80.4	76.5	77.1	74.5	71.6	81.7	75.2	71.3	78.8
SFA	78.8	75.8	75.7	81.3	75.6	76.9	75.4	72.4	82.6	74.7	72.4	80.7
DANN	80.5	77.6	78.8	83.2	76.4	77.2	77.6	73.5	84.2	75.1	74.3	82.2
AMN	84.5	81.2	82.7	85.6	82.4	81.7	81.7	76.6	85.7	81.5	80.9	86.1
HATN	86.6	86.3	87.4	86.5	85.7	86.8	84.3	81.5	87.9	84.7	84.1	87.0
IATN	87.0	86.9	85.8	86.8	86.5	85.9	84.1	81.8	88.7	84.4	84.7	87.6
BERT-DAAT	90.8	89.3	90.5	89.7	89.5	90.7	90.1	88.9	93.1	88.8	87.9	91.7
LSTM	75.6	73.4	-	78.6	75.2	-	72.2	69.6	-	-	-	-
TextGCN	80.8	77.6	79.2	85.3	81.1	79.7	82.6	78.2	82.3	83.3	84.1	81.7
FastGCN	81.6	80.6	81.1	86.0	82.7	82.0	83.5	78.7	84.5	84.2	85.7	83.4
GAST	87.9	87.3	89.1	88.2	86.2	87.4	85.6	83.4	89.3	87.7	87.5	89.4
BERT-GAST	**91.1**	**90.7**	**92.1**	**90.4**	**91.2**	**91.5**	**90.7**	**89.4**	**93.5**	**89.7**	**89.2**	**92.6**
G_*Non_Pos-Tran.*	85.9	84.7	87.6	86.8	83.4	85.5	84.2	80.4	87.8	85.8	85.5	87.4
G_*Non_HGAT*	86.6	85.9	88.1	87.4	85.0	86.1	84.5	81.3	88.2	86.4	86.7	88.2
G_*Non_IDS*	87.2	86.6	87.9	87.6	85.8	86.7	85.0	82.6	88.5	85.9	86.2	87.7
G_*Non_agg*	87.5	86.7	88.9	88.0	85.9	86.9	85.2	82.6	89.0	87.3	87.2	89.1
G_*Non_act*	87.3	86.3	88.7	87.7	85.3	86.2	84.8	81.8	88.7	86.9	87.1	88.7

the graph-based models (i.e., TextGCN and FastGCN) further highlights the superiority of GAST. We believe the reason is that GAST is able to consider the sequential information and syntactic structures jointly; hence, the comprehensive semantic features could be better encoded and learned.

- As Table 2 shows, the pre-trained language model (PLM, i.e., BERT-DAAT) can outperform all the existing CDSC methods and non-transfer approaches by significant margins, which proves the semantic extraction capabilities of the large-scale pre-trained language models in this task. Nevertheless, after incorporating the BERT embeddings, our proposed model (i.e., BERT-GAST) gets further improvement and achieves a new SOTA. It means that the performance of GAST can be further improved based on the advanced PLMs in the future.

In summary, all the evidence above indicate that GAST outperforms other strong baselines on diverse transfer tasks. These observations, meanwhile, imply that the domain-invariant semantics learned by the proposed model are more effective and transferable for cross-domain sentiment classification.

4.6 Ablation Study

In this subsection, we conduct multiple ablation studies to verify the effect of different modules. In what follows, we first describe the variants of GAST and then analyze how each of them affects the final performance:

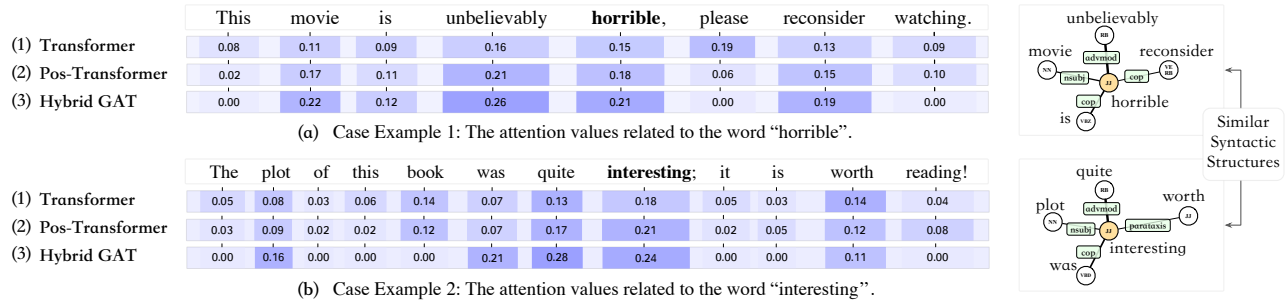
- **G_*Non_Pos-Transformer***: it utilizes the vanilla transformer instead of the POS-Transformer to learn the sequential feature representations, i.e., to eliminate the impact of POS-tag.
- **G_*Non_HGAT***: a variant of the proposed model which removes the HGAT module directly, i.e., to eliminate the effect of the syntactic dependency relations.
- **G_*Non_IDS***: a variant of the GAST model, which replaces integrated adaptive strategy with the classical GRL strategy[§], i.e., to verify the effectiveness of IDS.
- **G_*Non_agg*** and **G_*Non_act***: two different variants which remove the relation-aggregate function and relation-activation function respectively to verify the effect of them.

The ablation results are shown in Table 2. To be specific, from the comparison results of G_*Non_Pos-Transformer* and GAST, we can find that the performances decrease a lot when removing the POS-Transformer. It verifies the effectiveness of the POS-Transformer and demonstrates that the POS-tag information is significant. Besides, the results between G_*Non_HGAT* and GAST indicate that HGAT can encode the transferable relational features effectually. Moreover, the performance of *G_Non_IDS* falls far behind the standard GAST, which validates that our proposed IDS is more effective than the traditional GRL strategy. Finally, after eliminating the impact of relation-aggregate (G_*Non_agg*) and relation-activation (G_*Non_act*) function, the performance drops in varying degrees, which further validates the importance of modeling the syntactic relations and the effectiveness of our two relational functions. In conclusion, the above ablation results demonstrate that our GAST is able to facilitate performance through multiple modules and gains superior prediction improvement in the CDSC task.

4.7 Case Study

To intuitively assess the effects of syntactic information, we visualize the attention scores in different layers of two examples from the DVD and Book domain. As Figure 4 shows, the values in row (1) is attention score from the vanilla transformer, which means that

[§]GRL is a famous and widely used adversarial strategy in cross-domain sentiment classification task. For space saving, we detailed this strategy in the supplementary material (i.e., Appendix A).



(a) Case Example 1: The attention values related to the word "horrible".

(b) Case Example 2: The attention values related to the word "interesting".

Figure 4: Attention score visualization of the different words. The attention values from vanilla attention (i.e., $Att.(Q, K, V)$ in formula 2), POS-attention (i.e., $Att.(Q^t, K^t, V)$ in formula 2) and HGAT (i.e., β in formula 8) are associated with the row (1), row (2), and row (3) respectively in both examples. Note that, some values are infinitely close to 0. That makes sense because HGAT makes the attention value more concentrated on the syntactic-related words.

the calculation of each word's attention does not consider POS tags and relational information. Comparatively, the attention values in row (2) and row (3) assess the POS tags and the syntactic relations, respectively. In the right part of Figure 4, we also show the syntactic structures of those two examples w.r.t the sentiment words.

As we can see from example 1 in Figure 4, the vanilla transformer makes extra decisions on some unrelated word (e.g., *"this"*, *"please"*) and pays much attention to these uncritical words. On the contrary, POS-transformer can alleviate this problem by revising attention scores with the help of POS tags. We believe one of the reasons is that the POS tags (e.g., "JJ" in the right part of Figure 4) can enforce the model to pay more attention to sentiment-related words no matter in which domain. Besides, HGAT could deal with the problem more appropriately through the domain-invariant syntactic relations between words, i.e., highlight the crucial neighbor words (e.g., *"is"*, *"unbelievably"*) via dependency relations as shown in right part of Figure 4. Finally, we can also observe similar phenomenon in example 2, which indicates that syntactic features are invariant and transferable between domains, as we mentioned before.

To sum up, the above case examples' visualization results convince us that the domain-invariant syntactic information is essential for cross-domain sentiment classification. Meantime, our proposed GAST model can capture essential graph adaptive semantics that is reasonably necessary for domain adaptation.

4.8 Assessment of Adaptive Efficiency

To study the adaptive efficiency of different models, we test several models' performance on the target domain with varying training sample rates. From the overall results in Figure 5, we can observe some interesting phenomena. First, we find that GAST can be trained with only 10% samples, while IATN does not work well with such limited data. Meantime, GAST with 40% samples even performs better than IATN with 80% samples. All these observations denote that GAST gains an advanced adaptive ability and efficiency, thus significantly reducing the number of training samples. Second, BERT-GAST obtains superior results under all dataset scales, while the performance of other baseline models (i.e., BERT-DAAT, GAST, and IATN) is relatively low. This observation indicates that the representations learned from our BERT-GAST contain much more transferable sentiment knowledge than other baseline methods.

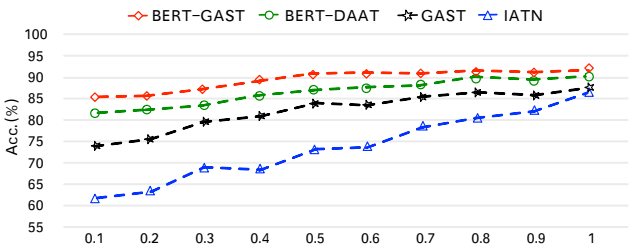


Figure 5: The influence of sample number. We explore the impact of sample number with different ratio (i.e., abscissa) of source domain. For the limited space, we only show the results of the task "$B \mapsto D$".

4.9 Influence of Adaptive Syntactic Graph

As we mentioned before, the syntactic structure plays a critical role in our proposed method. To evaluate the impact of different syntactic features, inspired by [40], we conduct a comparative experiment using two well-known dependency parsers, i.e., Stanford Parser [4] and Biaffine Parser[¶] [8], to construct our syntactic graph. The performance (i.e., accuracy, %) of these two category graphs on $\boldsymbol{D} \mapsto *$ tasks is shown in Table 3. The value in parentheses represents an absolute improvement, and method (1) indicates the G_*Non_HGAT* which is described in section 4.6. From the results, we can easily draw some conclusions. First, both two dependency parsers have made significant improvements compared with the method (1), the accuracy improvement on different transfer tasks is between 1.0% and 1.4%. Second, the quality of the syntactic graph does affect the final performance of the model, which indicates the effectiveness of syntactic features for CDSC research. Finally, it also implies that although existing parsers can capture most of the syntactic information correctly, our GAST's performance has potential to be further improved with the advances of parsing techniques in the future.

4.10 Hyper-parameter Study

As explained in previous studies [38, 39], multi-head attention can extract the relationship among words from multiple subspaces, thus it may increase the representation capability of our GAST model.

[¶]https://github.com/allenai/allennlp.

Table 3: The performance (%) of different syntactic graphs constructed by different parsers on $D \mapsto *$ tasks.

Syntax Parser	$D \mapsto B$	$D \mapsto E$	$D \mapsto K$
(1) Without Graph	86.6	85.9	88.1
(2) Stanford Graph	**87.1**	**86.6**	**88.6**
+compare with (1)	**(+0.5)**	**(+0.7)**	**(+0.5)**
(3) Biaffine Graph	**87.9**	**87.3**	**89.1**
+compare with (1)	**(+1.3)**	**(+1.4)**	**(+1.0)**
+compare with (2)	**(+0.8)**	**(+0.7)**	**(+0.5)**

Table 4: The Influence of model depth (i.e., attention heads) on $D \mapsto *$ tasks. The metric is accuracy (%).

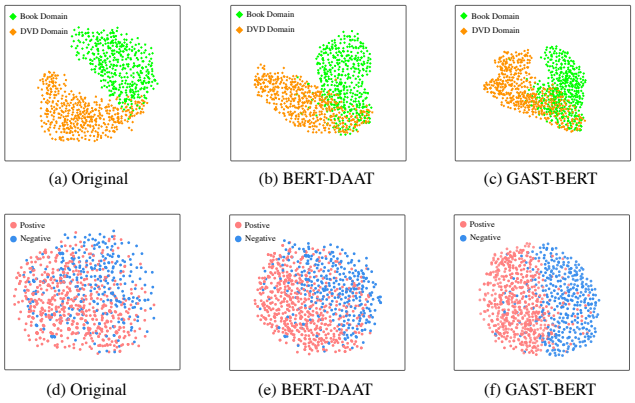
Models	$D \mapsto B$	$D \mapsto E$	$D \mapsto K$
HGAT w 1 head	86.9	86.4	87.5
HGAT w 2 head	87.2	86.8	88.4
HGAT w 3 head	**87.9**	**87.3**	**89.1**
HGAT w 4 head	87.7	87.2	88.7
HGAT w 5 head	87.5	86.9	88.2
Trans. w 5 head	86.6	86.1	88.4
Trans. w 6 head	87.6	86.7	88.7
Trans. w 7 head	87.5	87.0	89.1
Trans. w 8 head	**87.9**	**87.3**	**89.1**
Trans. w 9 head	86.8	87.1	88.8
Trans. w 10 head	87.2	87.3	89.0

We conduct experiments on the effect of multi-head attention by changing the number of attention heads. As the results shown in table 4, the best performance is achieved when we utilize three heads in HGAT and eight heads in POS-Transformer. With fewer heads, the performance drops at most 1.3% at $D \mapsto B$, 1.2% at $D \mapsto E$ and 1.6% at $D \mapsto K$, indicating the information from some crucial subspace is lost. However, even if we stack more heads, the performance also decreases at a similar level. It is likely due to the redundant feature interactions with more heads, which may weaken the performance of GAST as well.

4.11 Visualization of Adaptive Embedding

To further demonstrate the transferability of GAST, we visualize the learned feature embeddings of the original Glove, BERT-DAAT, and GAST-BERT (incorporating syntactic graph embeddings). Due to space constraints, we only show one transfer task (i.e., $B \mapsto D$). Specifically, in subfigure (a)~(c), we sample 500 reviews from domain B and 500 reviews from domain D. Moreover, we also sample 1,000 reviews (500 positive and 500 negative) randomly from the target domain D and project their feature embeddings via t-SNE [29].

As shown in Figure 6, from the original word representation (i.e., subfigure (a)) to final feature embeddings (i.e., subfigure (b) and (c)), the feature distributions between the source domain and target domains become more indistinguishable. The observation indicates that GAST-BERT can match the discrepancy between domain distribution and learn better domain-shared features than other methods (e.g., BERT-DAAT). Besides, the distribution of the original embedding is scattered and has a vague sentiment classification boundary. By contrast, the edge of GAST-BERT is more distinguishable than the original and BERT-DAAT. The reason may lie in that GAST-BERT can distill the domain-invariant and encode graphical sentiment knowledge so that the model can obtain a more discriminative sentiment feature in the target domain.



(a) Original (b) BERT-DAAT (c) GAST-BERT

(d) Original (e) BERT-DAAT (f) GAST-BERT

Figure 6: The t-SNE projection of the extracted features. The above three subfigures (i.e., (a)~(c)) show t-SNE visualization of different model's feature embedding for the $B \mapsto D$ task. The red and blue points in (d)~(f) denote the target positive and target negative examples, respectively.

In summary, the observations above indicate that our proposed GAST-BERT produces features that are easier to transfer across domains since it is domain awareness and helps to distill graphical feature embeddings from domain-invariant syntactic structures.

5 CONCLUSION

This paper presented a focused study on leveraging graph adaptive syntactic knowledge and sequential semantics for cross-domain sentiment classification. In particular, we proposed a novel domain adaptive method for CDSC called the Graph Adaptive Semantic Transfer (GAST) model, which mainly consists of two modules. The first is the POS-Transformer module, which is able to learn the overall semantic features from word sequences and POS tags. The second module is Hybrid Graph Attention (HGAT), designed to learn domain-invariant syntactic features from the syntactic graph. Moreover, the two modules are optimized by an integrated adaptive strategy, which ensures GAST understands invariant features better between domains. Experiments on four public datasets verified the effectiveness of our model. The ablation results and case studies further illustrate each module's point and explainability, which may provide insights for future work.

6 ACKNOWLEDGEMENTS

This research was partially supported by grants from the National Key Research and Development Program of China (Grant No. 2021YF F0901005), the National Natural Science Foundation of China (Grants No. 61922073, U20A20229 and 62006066), the Foundation of State Key Laboratory of Cognitive Intelligence, iFLYTEK, P.R.China (No. CI0S-2020SC05) and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR).

Appendix A

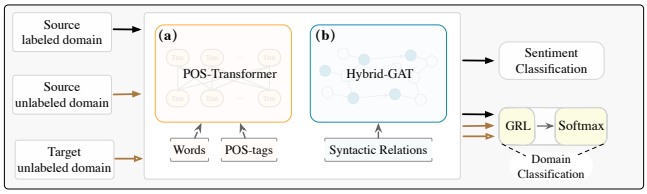


Figure 7: The framework of the ablation model G_*Non_IDS* as described in section 4.6. It mainly includes two tasks, i.e., sentiment classification and domain classification.

The proposed GAST model achieves sentiment domain adaptation by IDS, which aims to facilitate knowledge transfer across domains. However, to better evaluate the performance of IDS, we replace it with a widely used adaptive strategy (i.e., Gradient Reversal Layer, GRL [11]). Specifically, we treat the output of HGAT (i.e., H) as the final feature and feed into the *softmax* layer for domain classification. The goal of the domain classification task is to identify whether the example originates from source domain (i.e., $\mathcal{D}_s$) or target domain (i.e., $\mathcal{D}_t$). The formulation can be defined as:

$$\hat{y}^d = softmax(W_d R + b_d) . \tag{16}$$

The traditional training method is to minimize the classification error of the domain classifier so that the classifier can better distinguish the difference between domains. It means that minimize the domain classification loss will enforce the domain classifier to learn domain-specific features, which is contrary to our purpose(i.e., learn domain-invariant features). As mentioned before, our goal is to learn domain-invariant features that can not be discriminated between domains and thus we need to maximize the loss of the domain classifier. However, in this way, the training purpose of the sentiment classifier (i.e., minimize the sentiment classification error) and the domain classifier (i.e., maximize the domain classification error) will compete against each other, in an adversarial way. To eliminate this problem, we introduce a Gradient Reversal Layer (GRL) [9, 11, 25, 26, 47] to reverse the gradient during training progress so that both loss functions can be trained jointly. Formally, during the forward propagation, the GRL acts as an identity transformation $G(\cdot)$. During the back-propagation, the GRL takes the gradient from the subsequent level and changes its sign, i.e., multiplies it by -1, before passing to the preceding layer:

$$G(x) = x , \quad \frac{\partial G(x)}{\partial x} = -I . \tag{17}$$

Note that, the GRL ensures that the feature distributions over the two domains are more similar, thus resulting in the domain-invariant features. Through this way, the domain classifier can be trained by minimizing the *cross-entropy* for all data from the source domain $\mathcal{D}_s$ and the target domain $\mathcal{D}_t$:

$$L_d = -\frac{1}{N} \sum_{i=1}^{N} \left(y_d^i \ln \hat{y}_d^i + \left(1 - y_d^i\right) \ln \left(1 - \hat{y}_d^i\right) \right), \tag{18}$$

where y_d^i is the ground-truth of sample i. N is the sum of n_s from source domain and n_t from target domain. Due to space limitations, we omit the calculation process of GRL. For details, please refer to [11] or related work [9, 25, 26, 47].

REFERENCES

[1] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[2] John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*. 120–128.

[3] Yitao Cai and Xiaojun Wan. 2019. Multi-domain sentiment classification based on domain-aware embedding and attention. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 4904–4910.

[4] Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 740–750.

[5] Jie Chen, Tengfei Ma, and Cao Xiao. 2018. FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling. In *International Conference on Learning Representations*.

[6] Minmin Chen, Zhixiang Xu, Kilian Q Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*. 1627–1634.

[7] Ying Ding, Jianfei Yu, and Jing Jiang. 2017. Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[8] Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734* (2016).

[9] Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. 2020. Adversarial and domain-aware bert for cross-domain sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 4019–4028.

[10] Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495* (2014).

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[12] Deepanway Ghosal, Devamanyu Hazarika, Abhinaba Roy, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2020. KinGDOM: Knowledge-Guided DOMain adaptation for sentiment analysis. *arXiv preprint arXiv:2005.00791* (2020).

[13] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *International Conference on Machine learning (ICML)*. Omnipress, 513–520.

[14] Chenggong Gong, Jianfei Yu, and Rui Xia. 2020. Unified Feature and Instance Based Domain Adaptation for End-to-End Aspect-based Sentiment Analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7035–7045.

[15] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Adaptive Semi-supervised Learning for Cross-domain Sentiment Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[17] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 168–177.

[18] Binxuan Huang and Kathleen M Carley. 2019. Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 5469–5477.

[19] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186.

[20] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

[21] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised Machine Translation Using Monolingual Corpora Only. In *International Conference on Learning Representations*.

[22] Fangtao Li, Sinno Jialin Pan, Ou Jin, Qiang Yang, and Xiaoyan Zhu. 2012. Cross-domain co-extraction of sentiment and topic lexicons. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 410–419.

[23] Liang Li, Weirui Ye, Mingsheng Long, Yateng Tang, Jin Xu, and Jianmin Wang. 2020. Simultaneous learning of pivots and representations for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8220–8227.

[24] Tian Li, Xiang Chen, Shanghang Zhang, Zhen Dong, and Kurt Keutzer. 2020. Cross-Domain Sentiment Classification with In-Domain Contrastive Learning. *arXiv preprint arXiv:2012.02943* (2020).

[25] Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang. 2018. Hierarchical attention transfer network for cross-domain sentiment classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[26] Zheng Li, Yun Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-End Adversarial Memory Network for Cross-domain Sentiment Classification.. In *IJCAI*. 2237–2243.

[27] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5, 1 (2012), 1–167.

[28] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*. PMLR, 97–105.

[29] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[30] S Mahalakshmi and E Sivasankar. 2015. Cross domain sentiment analysis using different machine learning techniques. In *Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO-2015)*. Springer, 77–87.

[31] Lisa Meijer, Flavius Frasincar, and Maria Mihaela Truşcă. 2021. Explaining a neural attention model for aspect-based sentiment classification using diagnostic classification. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*. 821–827.

[32] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*. 751–760.

[33] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 79–86.

[34] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[35] Chuan Shi, Xiaotian Han, Li Song, Xiao Wang, Senzhang Wang, Junping Du, and S Yu Philip. 2019. Deep collaborative filtering with multi-aspect information in heterogeneous networks. *IEEE transactions on knowledge and data engineering* 33, 4 (2019), 1413–1425.

[36] Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5683–5692.

[37] Yuanhe Tian, Guimin Chen, and Yan Song. 2021. Aspect-based Sentiment Analysis with Type-aware Graph Convolutional Networks and Layer Ensemble. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2910–2922.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[40] Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[41] Wenya Wang and Sinno Jialin Pan. 2018. Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2171–2181.

[42] Rui Xia and Chengqing Zong. 2011. A POS-based ensemble model for cross-domain sentiment classification. In *Proceedings of 5th international joint conference on natural language processing*. 614–622.

[43] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7370–7377.

[44] Jianfei Yu and Jing Jiang. 2016. Learning Sentence Embeddings with Auxiliary Tasks for Cross-Domain Sentiment Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 236–246.

[45] Jianhua Yuan, Yanyan Zhao, Bing Qin, and Ting Liu. 2021. Learning to Share by Masking the Non-shared for Multi-domain Sentiment Classification. *arXiv preprint arXiv:2104.08480* (2021).

[46] Kai Zhang, Qi Liu, Hao Qian, Biao Xiang, Qing Cui, Jun Zhou, and Enhong Chen. 2021. EATN: An Efficient Adaptive Transfer Network for Aspect-level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[47] Kai Zhang, Hefu Zhang, Qi Liu, Hongke Zhao, Hengshu Zhu, and Enhong Chen. 2019. Interactive attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[48] Kai Zhang, Kun Zhang, Mengdi Zhang, Hongke Zhao, Qi Liu, Wei Wu, and Enhong Chen. 2022. Incorporating Dynamic Semantics into Pre-Trained Language Model for Aspect-based Sentiment Analysis. *arXiv preprint:2203.16369* (2022).

[49] Huang Zou, Xinhua Tang, Bin Xie, and Bing Liu. 2015. Sentiment classification using machine learning techniques with syntax features. In *2015 International Conference on Computational Science and Computational Intelligence*. 175–179.